IN THE SPECIFICATION:

Please amend the specification as follows.

Please replace the paragraph originally beginning on page 1, line 25 with the following paragraph:

A typical computer system has ~~I/O~~ input/output (I/O) access to one or more volumes on a storage device such as a hard drive. A volume generally includes an amount of free (unused) storage space which varies over time. When the number of computers and respective volumes in the networked system becomes very large, the total amount of unused storage may become significant.

Please replace the paragraph originally beginning on page 4, line 8 with the following paragraph:

In one embodiment, DCI may include ~~XML-capable~~ eXtensible Markup Language (XML) capable software applications on a peer-to-peer network. A source application on a first computer system may generate a message intended for a second computer system. Using DCI, the message may be translated from an original, internal, or other native format to a portable format (e.g., XML) on the first computer system, thereby generating a portable (e.g., XML) message. The portable message may include metadata which comprise identifying characteristics of the source application. The portable message may be sent from the first computer system to a second computer system using peer-to-peer message passing between the first computer system, the second computer system, and optionally one or more intermediary computer systems. After being received at the second computer system, the portable message may be routed using DCI to an appropriate target application based on the metadata.

Please replace the paragraph originally beginning on page 4, line 20 with the following paragraph:

In one embodiment, DCI may include small, network-unaware applications called "peerlets." Peerlets may be suitable for applications including chat, shared whiteboard, and other collaborative applications. A peerlet on a first computer system may generate a message (including collaborative data such as chat text or whiteboard graphics) and send the message to the distributed computing infrastructure using an ~~API~~ application programming interface (API). The DCI may translate the message from an original or native format to a portable format (e.g., XML), thereby generating a portable (e.g., XML) message, wherein the portable message comprises metadata which comprise identifying characteristics of the source peerlet. The portable message may then be sent from the first computer system to a second computer system using peer-to-peer message passing between the first computer system, the second computer system, and optionally one or more intermediary computer systems;. After the portable message is received at the second computer system, DCI may route the portable message to a target peerlet on the second computer system based on the metadata. Like the source peerlet, the target peerlet is configured to communicate using the API to the distributed computing infrastructure API.

Please replace the paragraph originally beginning on page 13, line 17 with the following paragraph:

Figures 1-3 illustrate computer system components that may be used in various embodiments of the invention. As Figure 1 indicates, in one embodiment, the system may include a plurality of computer systems, where each computer system may include at least one peripheral device, e.g., comprised in a human interface, and a computer blade. In one embodiment, a computer blade (or "blade") may comprise a "computer on a card." In other words, the computing system may be comprised on a circuit card which may include standard computing system components such as a ~~CPU~~ central processing unit (CPU), memory, power supply, and network interface, as well as an extender, e.g., a ~~PCI~~ Peripheral Component Interconnect (PCI) extender, for communicating with the remote human interface. Other types of computer systems and components, including standard

desktop PCs, may also be used to implement the system and method for virtual network attached storage. The computer system may include various components necessary for computer operations, such as, but not limited to, a processor and a storage medium. For further information regarding the use of multiple computer blades in a system, please see U.S. Patent Application Ser. No. 09/728,667 titled "Computer On A Card With A Remote Human Interface", which was filed December 12, 2000, whose inventors are Andrew Heller and Barry Thornton, which was incorporated by reference above.

Please replace the paragraph originally beginning on page 15, line 27 with the following paragraph:

As Figure 1 shows, connecting cables 151, 153, and 155 may connect computer blades 101, 105, and 109 to respective peripheral device groups through respective device ports or hubs, referred to herein as C-Ports, 157, 159, and 161. In one embodiment, each device port may comprise an extender device that may enable transmission of user interface signals (i.e., peripheral device signals) over distances generally not allowed by standard protocols such as ~~USB~~ Universal Serial Bus (USB). For further information regarding extended communications between a computer and a remote human interface, please see U.S. Patent Application Ser. No. 09/892,324 titled "Computer System Having a Remotely Located Human Interface Using Computer I/O Bus Extension", which was filed June 25, 2001, and U.S. Patent Application Ser. No. 09/892,331 titled "System Comprising Multiple Co-Located Computer Systems Each Having a Remotely Located Human Interface Using Computer I/O Bus Extension", both of which were incorporated by reference above.

Please replace the paragraph originally beginning on page 19, line 11 with the following paragraph:

Referring to Figure 3, an embodiment of a computer blade 105 having a power supply 210, hard drive 208, and motherboard 207 is shown. The computer blade 105 may include elements that make up a standard PC, such as, but not limited to, a

motherboard 207 with various components such as but not limited to a processor, e.g., a CPU 306, memory 304, and interface logic 302, which may include network logic 305, I/O logic 307, and human interface logic 303, as well as other interface circuitry associated with a motherboard 207, configured on a single card. The network logic 305 may include a LAN or WAN connection, such as but not limited to a IEEE803.2 (10/100 BaseT) Ethernet, and circuitry for connecting to peripheral devices coupled to the computer blade 105. The computer blade 105 may be electrically coupled to the cage 113 (shown in Figure 2) through the edge connector 209 or interfacing edge connector 309 that may face to the rear of the computer blade 105. In an embodiment of the invention, the computer blade 105 may slide into a slot 107 (shown in Figure 2) of the cage 113 (shown in Figure 2), making contact with the cage connector (not shown).

Please replace the paragraph originally beginning on page 27, line 24 with the following paragraph:

In one embodiment, the fail-over condition of computer blade 403 may be signaled manually, such as by a user calling a system administrator. In one embodiment, reconnecting a user's peripheral devices, e.g., keyboard 123 or 129 (see Figure 1), mouse 125 or 131, and monitor 127 or 133, may include identifying replacement computer blade 501, loading the failed computer blade 403 information onto the replacement computer blade 501 from either the first computer blade 401 or the third computer blade 405, and establishing a connection between the user's peripheral devices and the replacement computer blade 501, such as via a soft switch (not shown). In one embodiment, while the information is being restored to the replacement computer blade 501, information reads and information writes from and to the failed computer blade 403 may be diverted to the replacement computer blade 501 so that a user's productivity is not interrupted. In one embodiment, a replacement computer blade 501 may have the standard operating system and applications already stored on it. When a fail-over condition occurs with a user's computer blade, the peripheral device for the user's computer blade may be switched over to the replacement computer blade and the user may begin using the applications already stored on the replacement computer blade. Backup information may be restored

to the replacement computer blade in the background, and while the user uses applications already stored on the replacement computer blade, writes the user performs may be diverted to the replacement computer blade.

Please replace the paragraph originally beginning on page 36, line 1 with the following paragraph:

In one embodiment, the resource manager may operate to monitor resource usage for each of the plurality of computers. In other words, the resource manager may monitor performance metrics for each computer such as a total memory size, a used memory size, a virtual memory size, peripheral type, available ports, processor type, processor speed, type of installed applications, whether a user is logged in, frequency of login ins, percentage of usage of CPU, percentage of usage of hard disks, network hardware installed, network usage, usage of installed applications, video specifications, usage of CD-ROM, a variable imparted by the operating system, and a variable imparted by the ~~BIOS~~ Basic Input/Output System (BIOS), among others.

Please replace the paragraph originally beginning on page 41, line 15 with the following paragraph:

Referring to Figure 15, an embodiment of a memory stack 1501 for a computer blade storing information from other computer blades is shown. In one embodiment, the user's computer blade, e.g., computer blade 403, and two additional computer blades, e.g., computer blades 401 and 405, may each use memory space on the hard drive 208 in the user's computer blade 403. In the embodiment shown, the memory spaces used by the blades include memory spaces 1503, 1505, and 1507, although in other embodiments, other memory spaces may be defined and used. In addition, as Figure 15 indicates, there may be additional memory space 1509 available for use by a virtual network attached storage (VNAS) system 1509. In one embodiment, a storage network with a storage area network server may be coupled to the computer blade 401 and 405. The storage network server may make the storage medium of computer blade 401 accessible by the processor

of the computer blade 405, and to make the storage medium of the computer blade 405 accessible by the processor of the computer blade 401. In one embodiment, the organization and manipulation of the user's computer blade memory space may be such that the blade memory space does not have a single point of failure, as described below in detail. By eliminating single points of failure, the computer blades 401, 403, and 405 together may be more reliable for use in such applications as e-commerce, trading floors, and repair call centers, among others.

Please replace the paragraph originally beginning on page 42, line 13 with the following paragraph:

A fail-forward hard drive may also utilize ~~NAS/SAN~~ Network Attached Storage (NAS) and/or Storage Area Network (SAN) techniques. In one embodiment, the computer blades 401, 403, and 405 may operate as a distributed NAS server. For example, in one embodiment, the computer blades 401, 403, and 405 may utilize unused memory space in a manner analogous to that used by NAS and SAN, and may also track the location of hardware and information in the system. In one embodiment, a virtual NAS (VNAS) system may be implemented where the NAS server software is distributed across the peer computer blades 401, 403, and 405 (and/or other computer blades) in the network, thereby eliminating the NAS server as a point of failure. In one embodiment, each of the computer blades 401, 403, and 405 may maintain a copy of the NAS server software. In one embodiment, the computer blades 401, 403, and 405 may store the NAS server software and may be able to transfer a copy of the software to one of the remainder of the computer blades 401, 403, and 405 in the event of a failure of a computer blade 401, 403, or 405. As mentioned above, the computer blades 401, 403, and 405 may also use computer blades 401, 403, and 405 (i.e., each other) for other software storage, as desired. The VNAS system is described further with respect to Figures 24 and 25.

Please replace the paragraph originally beginning on page 50, line 16 with the following paragraph:

Figure 20 illustrates a swap move, according to one embodiment. As mentioned above, a swap move may be used to equalize or adjust the use of resources in a network (e.g., to put more demanding users with faster computer blades). The computer blades may be switched by two users, such as computer blades 2001 and 2003. Information, such as, but not limited to, applications and settings from one computer blade 2001, may be present on another computer blade 2003, post move, and vice-versa. In one embodiment, information from one of the computer blades 2005 and 2007 performing a switch, may be stored in a temporary third location to preserve the target computer blade 2007 while the switching computer blade 2005 overwrites the target computer blade's information. For example, an intermediate image server 2009 (based on ~~PXE~~ Preboot Execution Environment (PXE) technology) may be used. Large-scale moves may also be within the scope of the invention. In moving multiple computer blades, moves may be scheduled for Operating System settings, profiles, applications, and user information from old computer blades to new computer blades.

Please replace the paragraph originally beginning on page 52, line 27 with the following paragraph:

In one embodiment, DCI may include socket-level communications services provided through a multi-threaded server. In one embodiment, DCI may include embedded support for ~~HTTP~~ hypertext transport protocol (HTTP) communications provided through a multi-threaded server. In one embodiment, DCI may include embedded support for free-form XML communications provided through a multi-threaded server. In one embodiment, DCI may include ~~SOAP protocol~~ Simple Object Access Protocol or Service Oriented Architecture Protocol (SOAP) support.

Please replace the paragraph originally beginning on page 54, line 1 with the following paragraph:

DCI may provide a windowed, graphical interface that can help visualize computational results or monitor progress of multiple jobs in a single environment. In one

embodiment, applications may make use of the built in communications, directory, and XML routing capabilities of the underlying infrastructure. These applications would ordinarily be large and complex and would have to either utilize cryptic APIs such as ~~MPI~~ Message Passing Interface (MPI) or contain implementations of sophisticated message passing and communications technology. In one embodiment, DCI eliminates most of this development and management overhead and provides a simple and consistent environment to develop, deploy, and manage distributed or cluster capable applications.

Please replace the paragraph originally beginning on page 54, line 11 with the following paragraph:

Figure 22 is a block diagram illustrating a DCI architecture according to one embodiment. Each of two or more computer blades 101 (A and B, in this example) runs an operating system (OS) 2302. In one embodiment, the OS 2302 handles basic tasks like networking over ~~TCP/IP~~ Transmission Control Protocol/Internet Protocol (TCP/IP). Each DCI-enabled computer system on the network 115 may include a DCI stack. The DCI stack may include the core DCI framework 2304, one or more peerlet APIs 2306, and one or more peerlets 2308.

Please replace the paragraph originally beginning on page 56, line 4 with the following paragraph:

In 2409, a DCI "listener" in the core DCI framework on computer B may receive the XML message. In one embodiment, the DCI listener may utilize a ~~UDP~~ User Datagram Protocol (UDP) server to listen for incoming packets over an IP-based network connection. The use of UDP rather than TCP may allow for the rapid shipment of packets without the overhead of TCP. The UDP server may be multi-threaded for increased scalability and improved response time. In one embodiment, the actual communication between DCI-enabled computers may use a more reliable mechanism such as TCP.

Please replace the paragraph originally beginning on page 57, line 4 with the following paragraph:

In one embodiment, the VNAS architecture is based on a collection of "store nodes" which may include ordinary desktops or workstations. The store nodes contribute their excess disk capacity to the VNAS volume. Mount points, including computers such as desktop PCs, may allow a user to point to a single hostname (from a list of several) to "mount" and view the VNAS volume. In one embodiment, VNAS supports industry-standard protocols (e.g., ~~DAV~~ Distributed Authoring and Versioning (DAV)) which negate the need for any special software to be installed on a typical PC desktop in order for the user to browse the VNAS volume. There may be several mount points on a VNAS network, thereby eliminating a single point of failure.

Please replace the paragraph originally beginning on page 59, line 22 with the following paragraph:

Figure 26 is a screenshot that demonstrates the simple manner in which commands can be broadcasted to every node (Blade or PC) running the DCI platform. The dialog box 2650 on the upper right hand side allows commands and arguments to be entered, while the simple results screen 2660 on the left shows the output of the command as received from a particular node. With functionality of this sort, management tasks such as distributed process listing across multiple operating systems, process deletion, or invocation, may be easy to implement and use.

Please replace the paragraph originally beginning on page 60, line 2 with the following paragraph:

In one embodiment, the Autonomous Intelligent Management System (AIMS) includes a collection of agents, applications, and tools built on top of the Distributed Computing Infrastructure. AIMS may augment the capabilities of human ~~IT~~ Information

Technology (IT) resources and enable IT managers to easily manage much larger numbers of systems than would otherwise be possible. AIMS may use existing (i.e., logged) XML messages to "play back" tasks on DCI-enabled computers. AIMS may also use new (i.e., synthetic) XML messages to invoke functionality on DCI-enabled computers.